WE CLAIM:

## CLAIMS

1. In a wireless communications device, a method for organizing field upgradeable system software, the method comprising:

forming system software into a first plurality of symbol libraries, each symbol library comprising symbols having related functionality;

arranging the first plurality of symbol libraries into a second plurality of code sections; and,

executing the wireless device system software.

2. The method of claim 1 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes:

starting symbol libraries at the start of code sections;

the method further comprising:

storing the start of code sections at corresponding start addresses; and,

maintaining a code section address table cross-referencing code section identifiers with corresponding start addresses.

3. The method of claim 2 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging symbols to be offset from their respective code section start addresses; and

the method further comprising:

maintaining a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers.

4. The method of claim 1 wherein forming system software code into a first plurality of symbol libraries includes forming read-write data for the plurality of symbol libraries; and,

wherein arranging the first plurality symbol libraries into a second plurality of code sections includes arranging the read-write data in a shared read-write code section.

5. The method of claim 1 wherein forming system software code into a first plurality of symbol libraries includes forming a symbol accessor code;

wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the symbol accessor code in a first code section; and,

the method further comprising:

storing the symbol accessor code address in a first location in memory;

wherein executing the code sections as system software includes:

in response to referencing the first location in memory, accessing the symbol accessor code; and,

invoking the symbol accessor code to calculate the address of a sought symbol using a corresponding symbol identifier, and a corresponding code section identifier.

6. The method of claim 5 wherein invoking the symbol accessor code to calculate the address of the sought symbol includes accessing the code section address table and the symbol offset address table to calculate the address of the sought symbol.

7. The method of claim 5 wherein storing the symbol accessor code address in a first location in memory includes storing the symbol accessor code address in the first code section.

8. The method of claim 5 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging read-write data, the code section address table, the symbol offset address table, and the symbol accessor code in the first code section.

9. The method of claim 8 wherein executing the wireless device system software includes:

loading the read-write data, the code section address table, the symbol offset address table, the symbol accessor code, and the symbol accessor code address from the first code section into read-write volatile memory; and

accessing the read-write data, the code section address table, the symbol offset address table, the symbol accessor code, and the symbol accessor code address from read-write volatile memory.

10. The method of claim 2 wherein storing the start of code sections at corresponding start addresses includes:

creating a second plurality of contiguously addressed memory blocks;

identifying each memory block with a corresponding code section; and

storing code sections in the identified memory blocks.

11. The method of claim 10 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a third plurality of symbol libraries in a first code section;

wherein identifying each memory block with a corresponding code section includes identifying a first memory block with the first code section; and

wherein storing code sections in the identified memory blocks includes storing the third plurality of symbol libraries in the first memory block.

12. The method of claim 10 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a first symbol library in a first code section;

wherein identifying each memory block with a corresponding code section includes identifying a first memory block with the first code section; and

wherein storing code sections in the identified memory blocks includes storing the first symbol library in the first memory block.

13. The method of claim 10 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes sizing the code sections to accommodate arranged symbol libraries; and,

wherein creating a second plurality of contiguously addressed memory blocks includes sizing memory blocks to accommodate corresponding code sections.

14. The method of claim 13 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes sizing the code sections to accommodate sizes larger than the arranged symbol libraries.

15. In a wireless communications device, a method for organizing field upgradeable system software, the method comprising:

forming system software code into a first plurality of symbol libraries, each library comprising symbols having related functionality;

arranging the first plurality of symbol libraries into a second plurality of code sections so that symbol libraries start at the start of code sections;

creating a second plurality of contiguously addressed memory blocks;

identifying each memory block with a corresponding code section;

storing code sections in the identified memory blocks, with the start of code sections at corresponding start addresses;

maintaining a code section address table cross-referencing code section identifiers with corresponding start addresses; and,

executing the wireless device system software.

16. The method of claim 15 wherein forming system software code into a first plurality of symbol libraries includes forming a symbol accessor code;

wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging symbols to be offset from their respective code section start addresses, and includes arranging the symbol accessor code in a first code section; and

the method further comprising:

maintaining a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers; and,

storing the symbol accessor code address in a first location in memory.

17. In a wireless communications device, a field upgradeable system software structure, the system software structure comprising:

executable system software differentiated into a second plurality of code sections; and,

a first plurality of symbol libraries arranged into the second plurality of code sections, each library comprising symbols having related functionality.

18. The system software structure of claim 17 wherein symbol libraries are arranged to start at the start of code sections; and,

the system software structure further comprising:

a memory with the start of code sections stored at corresponding start addresses; and,

a code section address table cross-referencing code section identifiers with corresponding code section start addresses in memory.

19. The system software structure of claim 18 wherein symbols are arranged to be offset from respective code section start addresses; and

the system software structure further comprising:

a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers.

20. The system software structure of claim 17 wherein the second plurality of code sections includes a shared read-write section; and,

wherein the first plurality of symbol libraries includes read-write data arranged in the read-write section.

21. The system software structure of claim 17 wherein the second plurality of code sections includes a first code section;

wherein the first plurality of symbol libraries includes a symbol accessor code arranged in the first code section to calculate the address of a sought symbol;

wherein the memory includes the symbol accessor code stored at an address; and,

the system software structure further comprising:

a first location for storage of the symbol accessor code address.

22. The system software structure of claim 21 wherein the symbol accessor code accesses the code section address table and the symbol offset address table to calculate the address of the sought symbol.

23. The system software structure of claim 22 wherein the first location is in the first code section.

24. The system software structure of claim 22 wherein the first plurality of symbol libraries includes read-write data, the code

section address table, the symbol offset address table, and the symbol accessor code arranged in the first code section.

25. The system software structure of claim 24 further comprising:

a read-write volatile memory; and,

wherein the read-write data, code section address table, the symbol offset address table, the symbol accessor code, and the symbol accessor code address are loaded into the read-write volatile memory from the first code section for access during execution of the system software.

26. The system software structure of claim 17 wherein the memory includes a second plurality of contiguously addressed memory blocks, each memory block storing a corresponding code section from the second plurality of code sections.

27. The system software structure of claim 26 wherein the second plurality of code sections includes a first code section;

wherein the first plurality of symbol libraries includes a third plurality of symbol libraries arranged in the first code section;

wherein the memory includes a first memory block to store the first code section; and,

wherein the first memory block stores the third plurality of symbol libraries.

28. The system software structure of claim 26 wherein the second plurality of code sections includes a first code section;

wherein the first plurality of symbol libraries includes a first symbol library arranged in the first code section;

wherein the memory includes a first memory block to store the first code section; and,

wherein the first memory block stores the first symbol library.

29. The system software structure of claim 26 wherein each of the second plurality of code sections has a size in bytes that accommodates the arranged symbol libraries; and,

wherein each of the contiguously addressed memory blocks have a size in bytes that accommodates corresponding code sections.

30. The system software structure of claim 29 wherein each of the second plurality of code sections has a size larger than the size needed to accommodate the arranged symbol libraries.

31. The system software structure of claim 17 wherein the memory is a writeable, nonvolatile memory.

32. In a wireless communications device, a field upgradeable system software structure, the system software structure comprising:

executable system software differentiated into a second plurality of code sections;

a first plurality of symbol libraries arranged at the start of the second plurality of code sections, each library comprising symbols having related functionality;

a memory including a second plurality of contiguously addressed memory blocks, each memory block storing a corresponding code section, from the second plurality of code sections, at a start address;

a code section address table cross-referencing code section identifiers with corresponding code section start addresses in memory; and,

a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses in memory, and corresponding code section address identifiers.